

Bioinformatic approaches to regulatory genomics and epigenomics

376-1347-00L | week 03

Pierre-Luc Germain

Plan for today

- Debriefing on the assignments
- Overview of NGS technologies, ChIP-seq and its analysis
- Practical:
 - primary processing of a ChIP-seq experiment
(to be continued next week)

Debriefing on the assignments

- Handing in the exercises etc.:

- Handing in the exercises: Please name the exercises files just **assignment.html** !!!
- Sync fork before committing:

This branch is 3 commits behind ETHZ-INS:main.

 Contribute ▾

 Sync fork ▾

- Please join the help channel for hints & questions concerning the exercises, git, package installation etc.
- Use titles and subtitles with **#/##** for the separate questions. Makes the documents structured and easier to read

1. AnnotationHub

a) Mouse EnsDb, v102, GRCm38

Debriefing on the assignments

- Exercise 2
 - **Gene ids** vs **gene names** vs **gene symbols**
 - **gene ids**: stable ID from Ensembl, truly unique, e.g. "ENSMUSG00000005677"
 - **gene symbols**: HGNC symbol from the [HUGO Gene Nomenclature Committee](#)
 - variable naming: `exons_mouse <- width(exsPerTx)`
 - more fitting would be: `exon_widths <- width(exsPerTx)`
 - `genes(ensdb, filter = TxBiotypeFilter("protein_coding"))`
 - here one would use as a filter: `GeneBioTypeFilter("protein_coding")`

Debriefing on the assignments

- Exercise 2

- If several annotation/sequences are obtained for query one can look at the metadata with:

```
colnames(mcols(q))
```

```
## [1] "title"          "dataprotider"    "species"
## [4] "taxonomyid"     "genome"          "description"
## [7] "coordinate_1_based" "maintainer"      "rdatadateadded"
## [10] "preparerclass"  "tags"            "rdaclass"
## [13] "rdatapath"      "sourceurl"       "sourcetype"
```

```
date_added <- mcols(q)[,c("rdatadateadded", "genome")]
date_added[order(date_added$rdatadateadded),]
```

```
## DataFrame with 19 rows and 2 columns
##      rdatadateadded      genome
##      <character> <character>
## AH49775      2015-12-28      GRCm38
## AH50120      2015-12-29      GRCm38
## AH50611      2016-05-03      GRCm38
## AH51299      2016-08-15      GRCm38
## AH51645      2016-11-03      GRCm38
## ...          ...          ...
## AH70177      2019-04-29      GRCm38.p6
## AH77927      2019-10-29      GRCm38.p6
## AH82549      2020-04-27      GRCm38.p6
## AH84787      2020-10-26      GRCm38.p6
## AH88477      2020-10-27      GRCm38.p6
```

Debriefing on the assignments

- Exercise 2
 - We can get the exons per transcript in the following way:

```
exs <- exonsBy(ensdb,  
               column=c("tx_id", "tx_biotype"),  
               filter=TxBiotypeFilter("protein_coding"))
```

Debriefing on the assignments

- Exercise 2
 - Calculating the number of exons per transcripts

```
# calculating the number of exons per transcript
nbExonsPerPCtx <- lengths(exs)
hist(nbExonsPerPCtx)
```

- Calculating the lengths of (spliced transcripts) using `width()`

```
# with width we can get the lengths of all exons per transcript in a list
ew <- width(exs)

# by summing the exon lengths per transcript we get the spliced transcript lengths
tl <- sum(ew)
|
# Plot
hist(tl, breaks=100)
```

Debriefing on the assignments

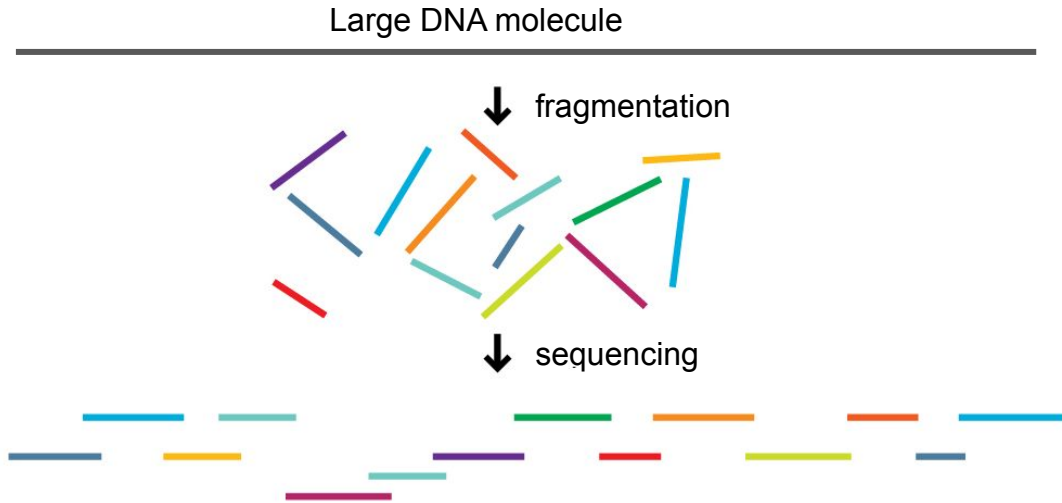
- Exercise 2
 - Alternatively: Calculating the lengths of (spliced transcripts) using `lengths()`

```
# with width we can get the lengths of all exons per transcript in a list  
ew <- lapply(exs, lengths)
```

```
# by summing the exon lengths per transcript we get the spliced transcript lengths  
tl <- lapply(ew, sum)  
length(tl)
```

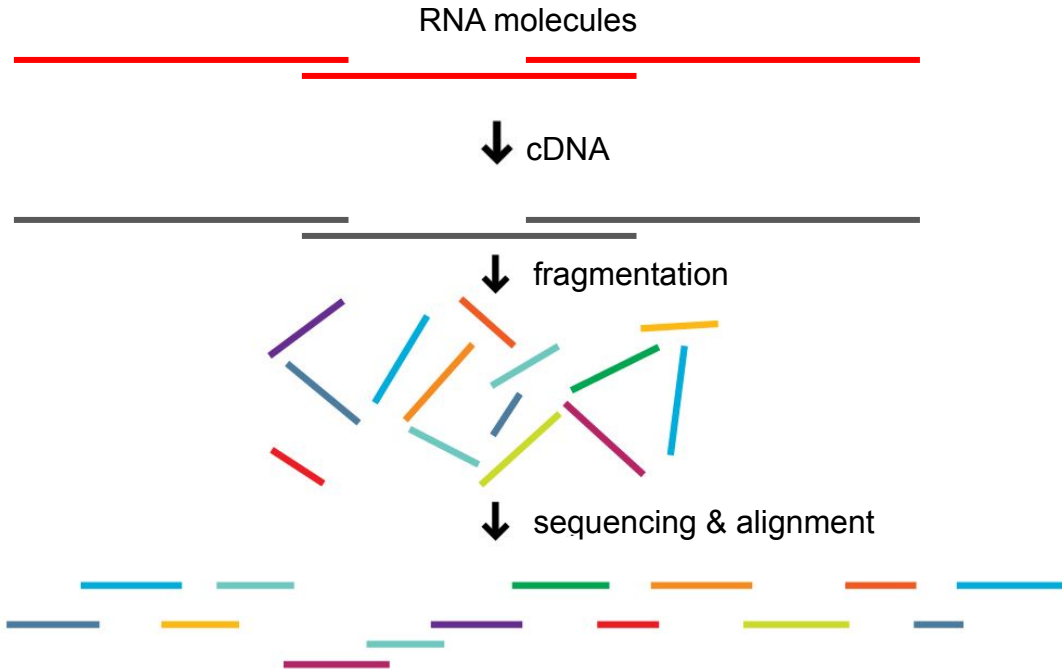

Next Generation Sequencing (NGS)

Shotgun sequencing:



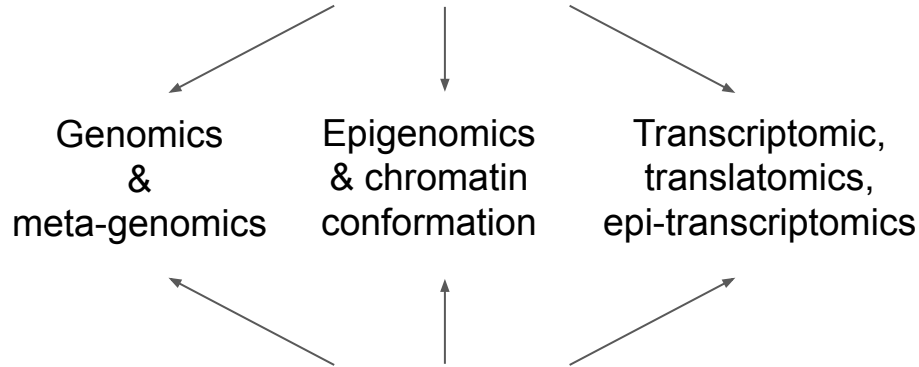
Next Generation Sequencing (NGS)

RNA sequencing:

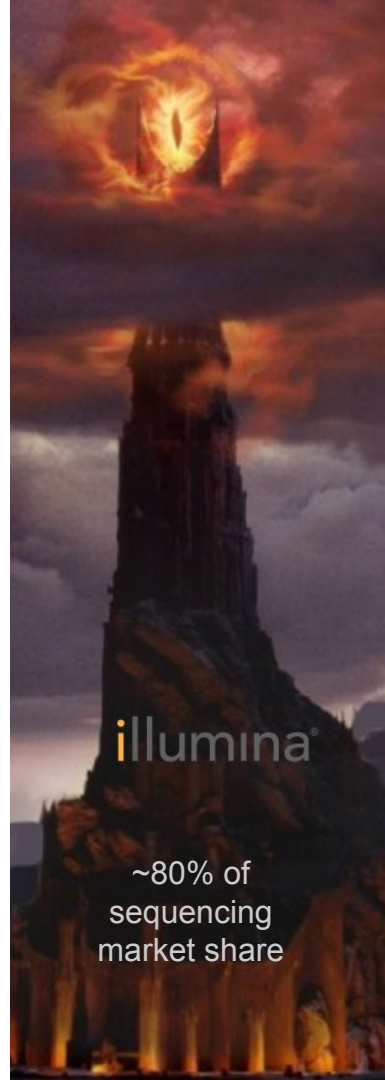




Next Generation Sequencing: one technology to rule them all

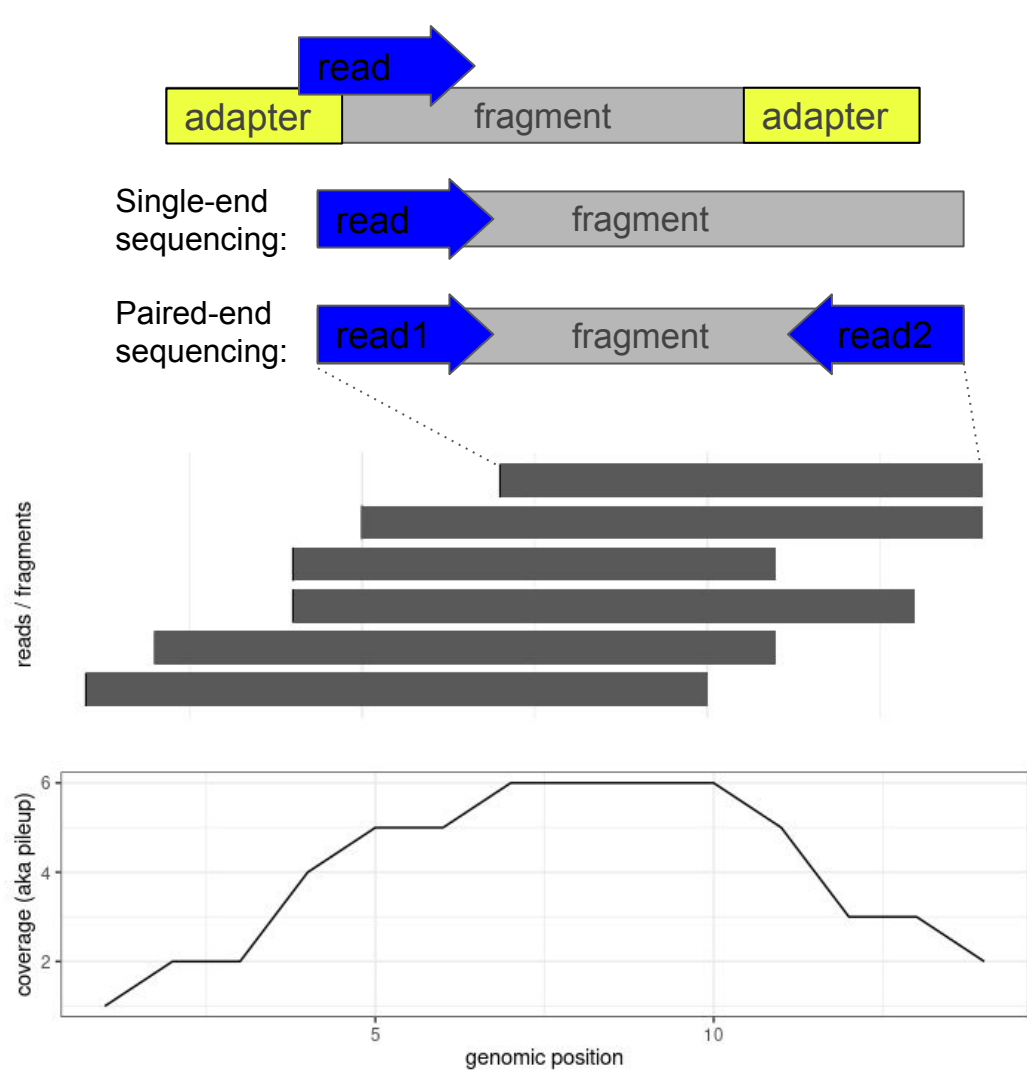
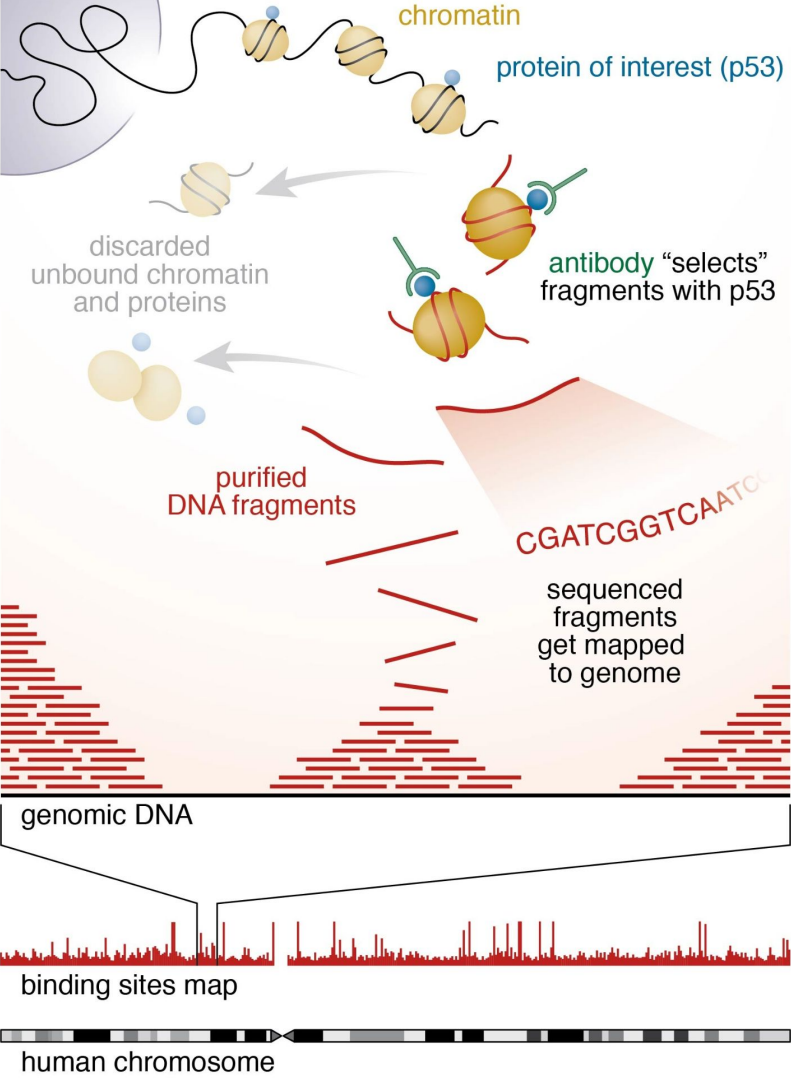


A lot of convergence in terms of analysis
tools and techniques

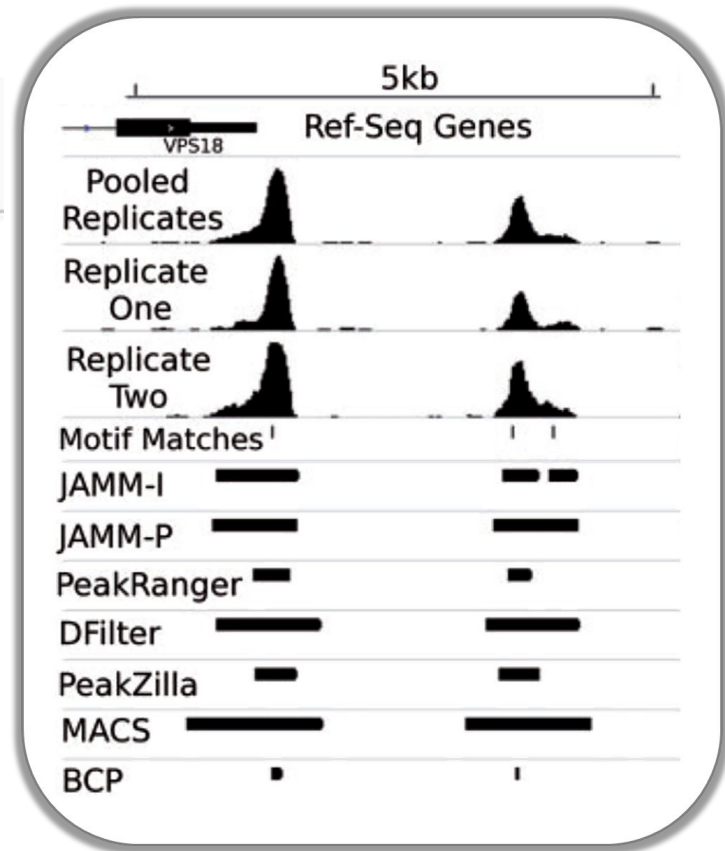
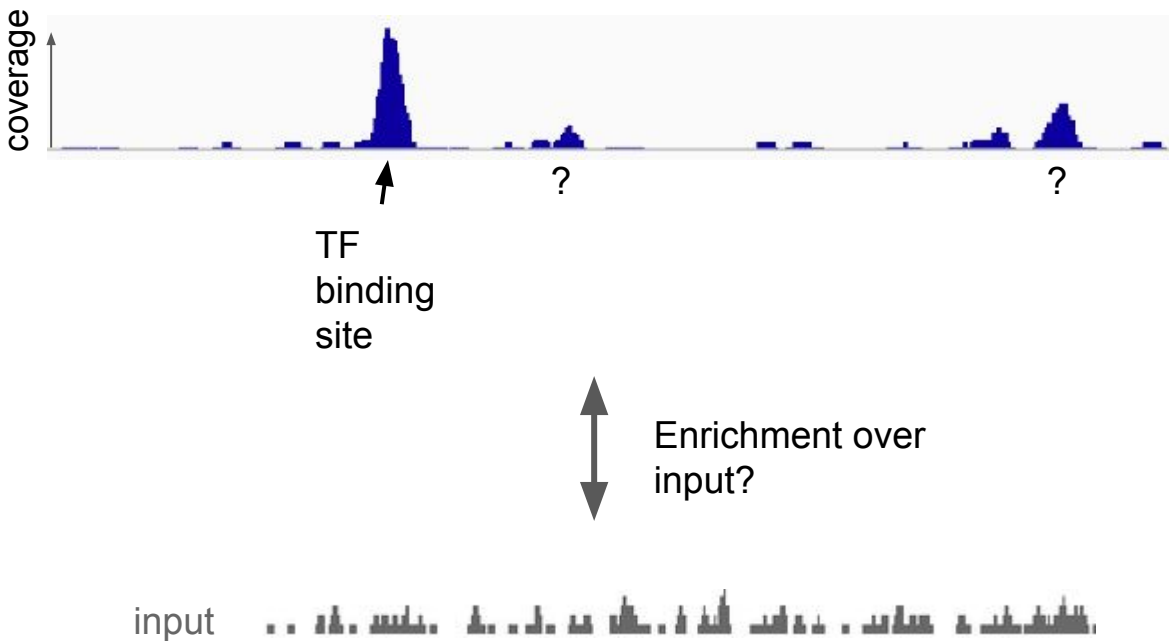


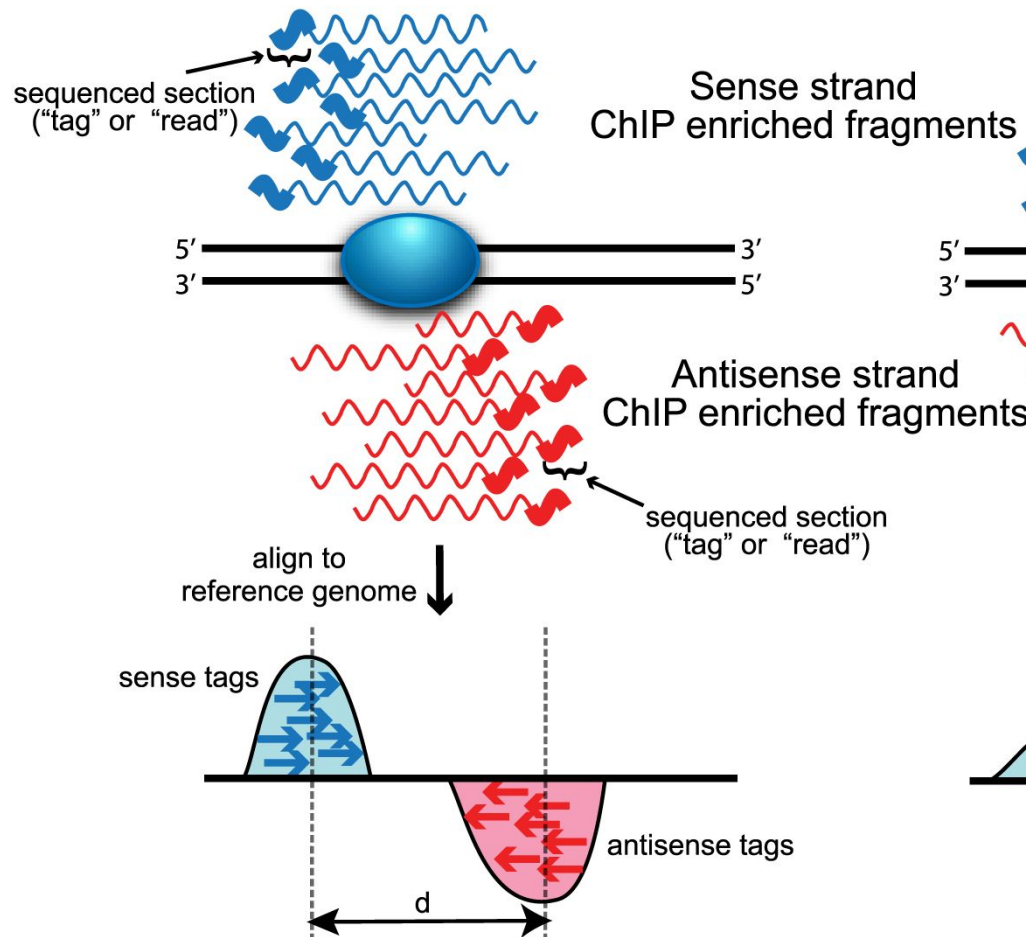
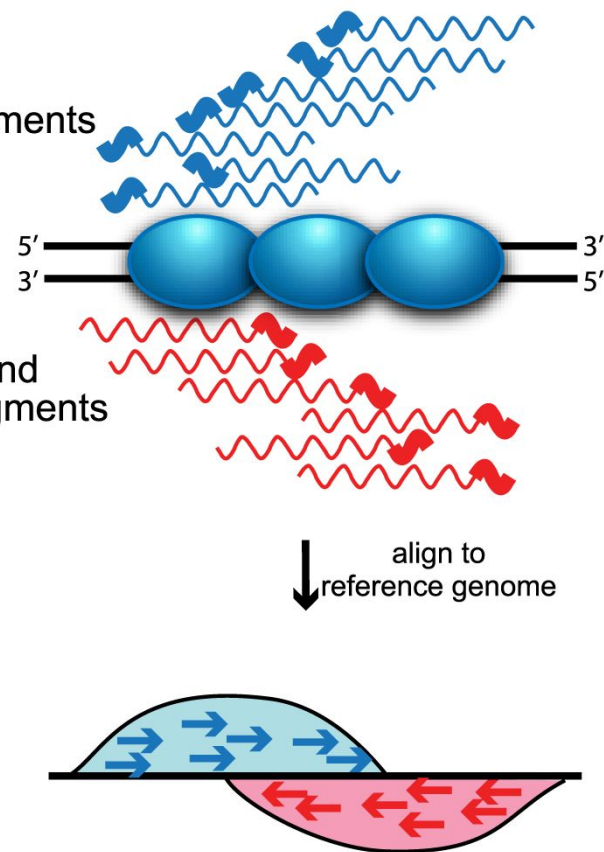
illumina®

~80% of
sequencing
market share

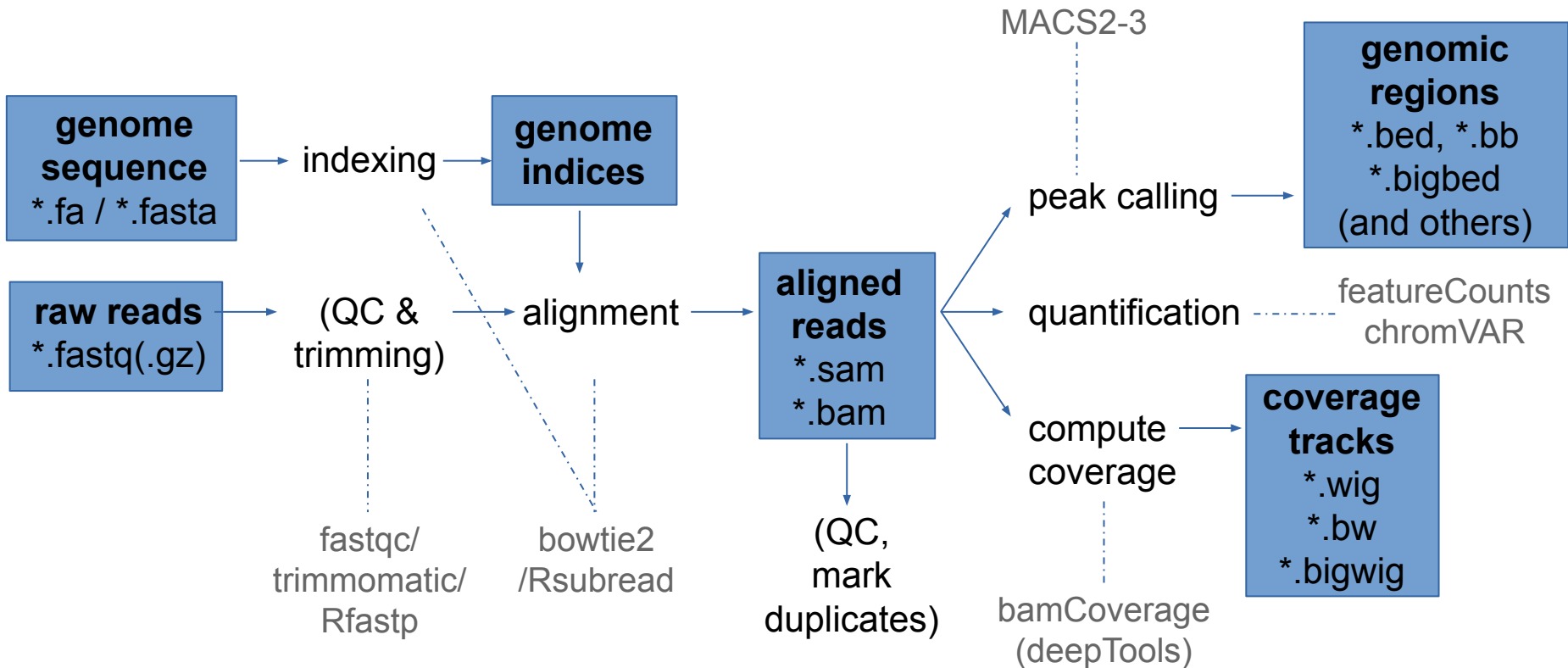


Peak calling



A**B**

Overview of a primary analysis pipeline (ChIP-seq and the likes)



Alternative toolsets for (DNA) primary analysis

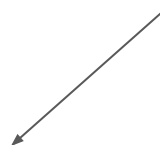
- The most standard one:

- [fastqc](#)
- [trimmomatic](#)
- [bowtie2](#)
- [picard](#)
- [deeptools](#)

- Pure R-based

- [rfastp](#)
- [Rsubread](#)

[QuasR](#)

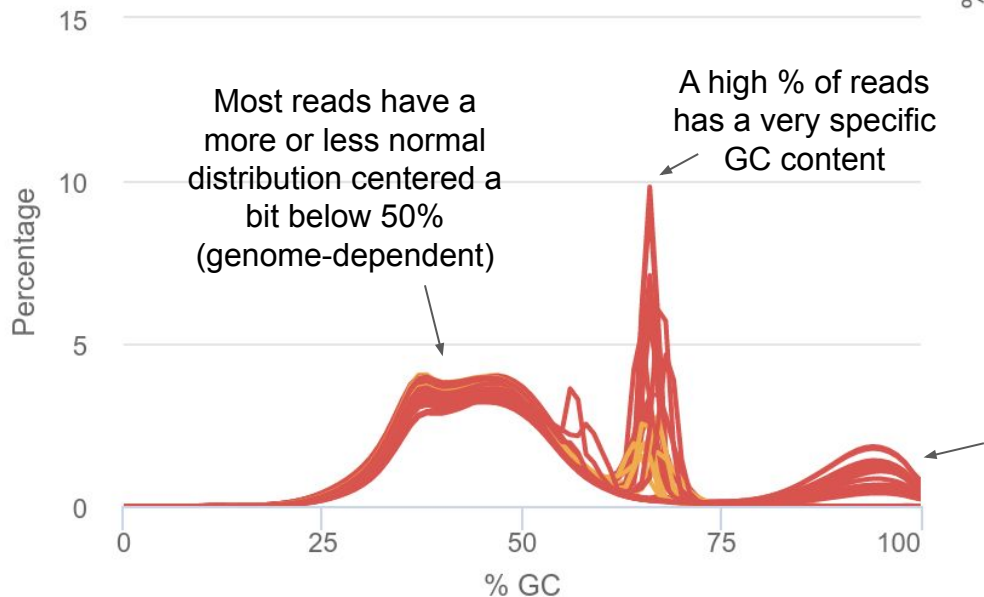


Downstream analysis (R)

- [epiwraps](#)
- [ChIPseeker](#)
- etc...

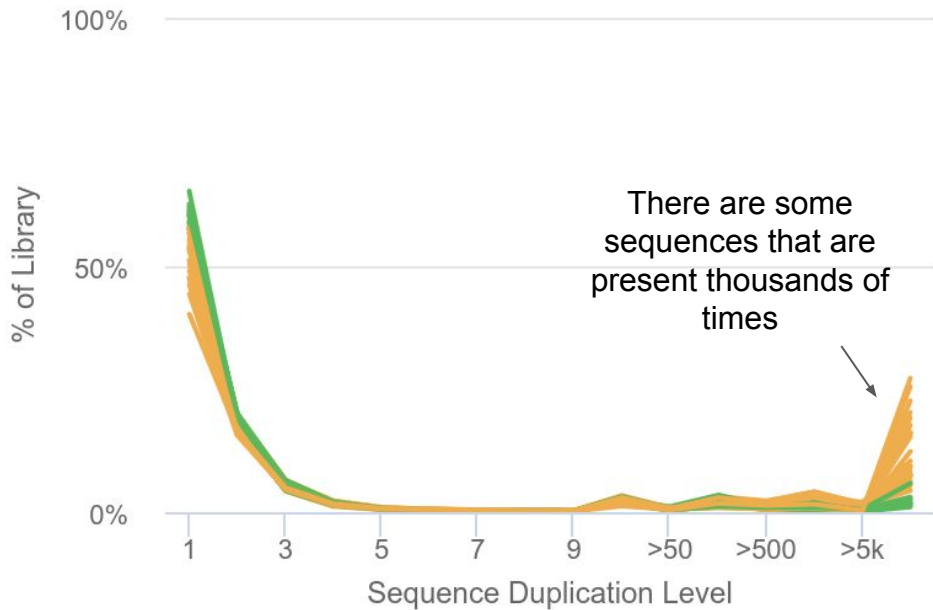
Example (rather extreme) QC problems

FastQC: Per Sequence GC Content



Created with MultiQC

FastQC: Sequence Duplication Levels

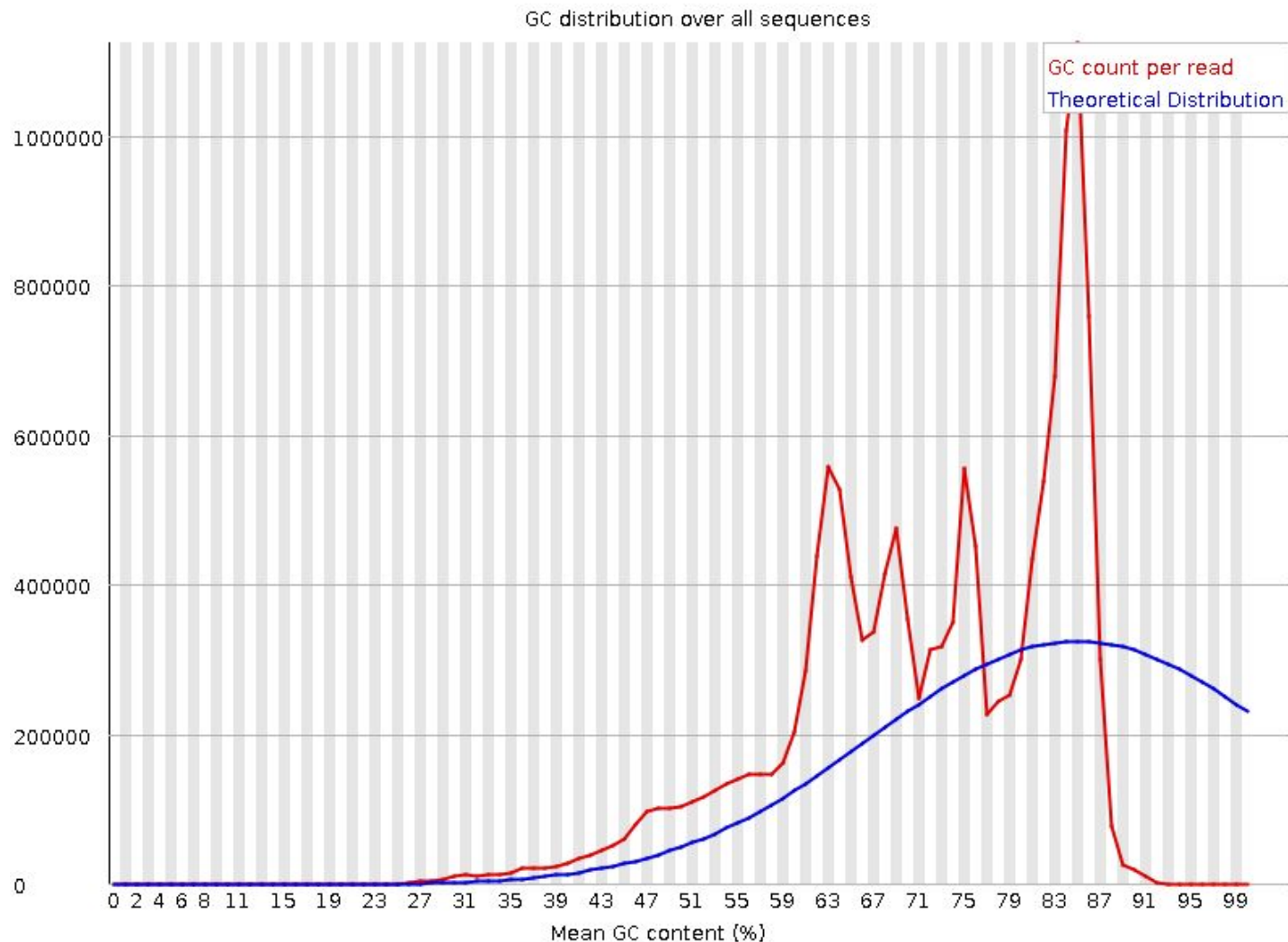


Created with MultiQC

Example (rather extreme)

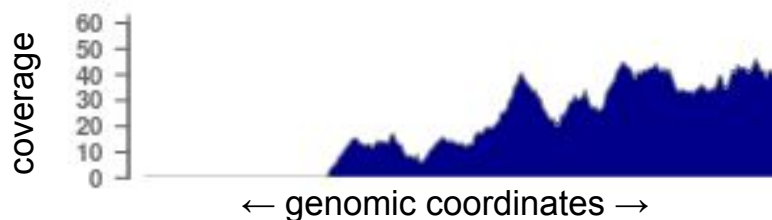
QC problems:

Bias from overamplification



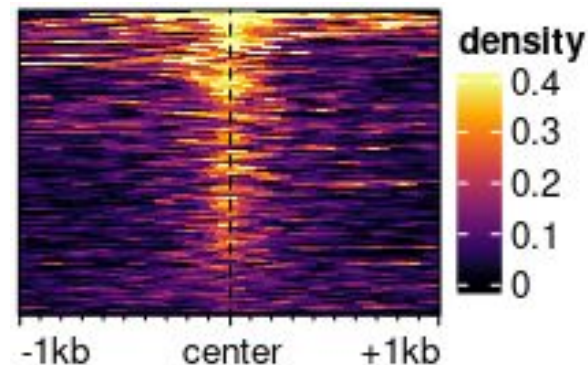
Visualizations available in *epiwraps*

- Signal across one genomic region:
`plotSignalTracks`



(Based on the *Gviz* R package)

- Signal across several genomic regions:
`signal2Matrix` →
`plotEnrichedHeatmaps`



(Mainly based on the *EnrichedHeatmap* R package, itself based on *ComplexHeatmap*)

Assignment

- Download the following Drosophila ChIP-seq for the protein CTCF:
 - IP: <https://www.encodeproject.org/files/ENCFF127RRR/@@download/ENCFF127RRR.fastq.gz>

(no input control for the purpose of this exercise)
- Process it from the raw data, obtaining:
 - bam file
 - peaks
- Report:
 - how many reads (and what percentage) were mapped
 - how many peaks were found
- Plot the signal around one of the peaks
- Please make sure that you name your final file **assignment.html** !!